NASA Contractor Report 201659

ICASE Report No. 97-12

A NONLINEAR PROGRAMMING PERSPECTIVE ON SENSITIVITY CALCULATIONS FOR SYSTEMS GOVERNED BY STATE EQUATIONS

Robert Michael Lewis

NASA Contract No. NAS1-19480
February 1997

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA 23681-0001

Operated by Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-0001

A nonlinear programming perspective on sensitivity calculations for systems governed by state equations

Robert Michael Lewis *
ICASE
Mail Stop 403
NASA Langley Research Center
Hampton, VA 23681-0001
buckaroo@icase.edu

Abstract

This paper discusses the calculation of sensitivities, or derivatives, for optimization problems involving systems governed by differential equations and other state relations. The subject is examined from the point of view of nonlinear programming, beginning with the analytical structure of the first and second derivatives associated with such problems and the relation of these derivatives to implicit differentiation and equality constrained optimization. We also outline an error analysis of the analytical formulae and compare the results with similar results for finite-difference estimates of derivatives. We then attend to an investigation of the nature of the adjoint method and the adjoint equations and their relation to directions of steepest descent. We illustrate the points discussed with an optimization problem in which the variables are the coefficients in a differential operator.

*This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001. This work was also supported by the State of Texas under the Geophysical Parallel Computation Project, contract 1059.

1 Introduction

This paper discusses the calculation of sensitivities, or derivatives, for optimization problems governed by ODE, PDE, and other state equations. The context for this discussion is the general nonlinear programming problem

$$\begin{array}{ll} \text{minimize} & F(a) = f(a, u(a)) \\ \text{subject to} & C_E(a, u(a)) = 0 \\ & C_I(a, u(a)) \geq 0, \end{array} \tag{1}$$

with the distinguishing feature that $u(a)$ is the solution of some set of equations,

$$h(a, u(a)) = 0. \tag{2}$$

For instance, (2) might represent the solution of the boundary-value problem

$$\begin{array}{c} -(K_a(x)u'(x))' = q(x), \quad x \in [0,1] \\ u(0) = u(1) = 0 \end{array} \tag{3}$$

where the coefficient $K_a(x)$ is given, say, by

$$K_a(x) = \sum_{i=1}^{n} a_i \phi_i(x)$$

for some fixed set of functions $\phi_1, \cdots, \phi_n$.

While our discussion will focus on the case where the equations defining u are differential equations, other defining relations are possible. Problems of the form (1)–(2) can appear in discrete event simulation. Another example is the sensitivity of eigenvalues and eigenvectors. For instance, if $A = A(a)$ is a smooth, $n \times n$ symmetric matrix-valued function of a, the system

$$\begin{aligned} Av - \lambda v &= 0 \\ v^T v - 1 &= 0 \end{aligned}$$

defines an eigenvalue-eigendirection pair $u = (\lambda, v)$. The pair (λ, v) is a smooth function of a when λ is a simple eigenvalue, and one can apply the formulae we discuss here to compute the related sensitivities.

The equation (2) typically describes the physical state of the problem. Examples of optimization problems governed by state relations abound in inverse problems, parameter estimation, remote sensing, optimal design, and optimal control. We will refer to the variable a as the *model parameters* and to $u(a)$ as the *state* associated with a. The governing equation (2) will be called the *state equation*.

We will examine the calculation of the derivatives associated with the problem (1). We will henceforth ignore the constraints C_E and C_I in (1) and consider the ostensibly unconstrained problem

$$\text{minimize} \quad F(a) = f(a, u(a)) \tag{4}$$

and study the derivatives of F and u with respect to the variable a, since the derivatives of C_E and C_I with respect to a are similar to those of F. This simplification helps us focus on the salient feature of $u(a)$: its nature as the solution of (2).

Our discussion of the calculation of sensitivities is motivated primarily by an interest in applying nonlinear programming algorithms to (1). The most generally effective optimization algorithms for problems such as these are quasi-Newton methods [8, 11], which require derivatives of the the objective function F and the constraints. Sensitivities are also useful in their own right to study the dependence of the state, objective, or constraints on the parameters a. As we shall see, the governing equation (2) imparts a great deal of structure to the calculation of derivatives.

The goal of this paper is to interpret the language that one encounters in the literature on calculating sensitivities for differential equations in more familiar terms, and, in particular, to show the connections to classical ideas in nonlinear programming. Because we have in mind the optimization of systems governed by differential equations, we will frame our discussion in the general terms of functional analysis.

The main theme of this paper is the systematic approach to computing derivatives based on implicit differentiation, and the significance of these derivatives for optimization. Among the particular points we will discuss are the following:

- A careful derivation of the general formulae for the first and second derivatives of F, including the infinite-dimensional case.
- The connection between the formulae for derivatives and equality constrained optimization.
- A comparison of numerical error estimates for sensitivity calculations via analytical formulae and by finite-differences.
- The distinction between the derivative and directions of steepest descent.
- The adjoint approach, and the sense in which the "adjoint equations" are adjoint.
- Some potential difficulties with the adjoint approach in the context of optimization algorithms; in particular, how it may correspond to a non-standard choice of scaling for some problems.

This exposition is intended partly as a primer for those unfamiliar with this type of sensitivity calculation, and partly to make it easier for those whose background lies in differential equations and those whose background lies in nonlinear programming to discuss optimization problems of mutual interest.

The problem that is tacitly assumed as the model problem in this paper is the case where $u(a)$ is the solution of a differential equation and a represents

either data in the problem—boundary values and source terms—or coefficients in the differential operator. Such problems make up a large proportion of those encountered in control and parameter estimation. One topic that we will not discuss is shape optimization, in which the domain on which the state is defined varies, since this topic requires a great deal of machinery unlike that developed here. However, many shape optimization problems can be reduced to problems where the domain of definition for u is fixed and the variation in the shape is represented by the variation of some boundary term or coefficient, in which case the approach discussed here applies. For examples, see [23].

We begin in §2 with the derivation of formulae for derivatives. The results in this section are certainly not new, but the emphasis placed on the role of implicit differentiation may be unfamiliar to some, and the presentation is detailed, particularly the general derivation and interpretation of the reduced Hessian, which relies on various technical identifications relegated to §11.

In §4 we present some error analysis for the calculation of sensitivities via the analytical formulae and compare the results with similar results for finite-difference estimates of derivatives. This comparison helps explain the often noted experience that analytical derivatives can be much more accurate than finite-difference approximations of sensitivities for systems governed by state equations.

In §5, we discuss the relationship between the formulae for derivatives and equality constrained optimization. Here we examine what is called the adjoint state or costate in the differential equations and control literature and identify it as a familiar Lagrange multiplier estimate in linear and nonlinear programming.

In §6 and §7 we discuss two approaches to sensitivity calculations. In practice, these approaches differ in the way in which they organize the intermediate calculations. The first is the sensitivity equations approach, which yields directional derivatives. The second is the adjoint equations approach, which is an attempt to represent the derivative in a particular form and obtain a direction of steepest descent by inspection. Our discussion is based on the distinction between the derivative, which is a linear functional and as such lives in the dual of the domain on which the problem is posed, and directions of steepest descent, which are vectors in the domain that depend on a choice of norm. In $I\!R^n$ linear functionals are simply row vectors that may be transposed to obtain a direction of steepest descent. However, in the infinite-dimensional case the situation is more complicated. This we also discuss in §7, where we clarify what is "adjoint" about the adjoint equations in the context of optimization, and how the adjoint equations are related to a choice of norm, or scaling, defining a direction of steepest descent.

We illustrate the discussion with a parameter estimation problem for an elliptic operator in §3 and §8. This example suffices to show how one computes first and second derivatives and directions of steepest descent with respect to different norms. This example also shows how one can go wrong by an uncritical use of the adjoint equations when they correspond to an unsuitable scaling for

the problem.

2 Formulae for the derivatives

We begin with the analytical formulae for derivatives for problems governed by state equations. These derivatives of the state and objective will follow from implicit differentiation. These formulae are derived in detail in order to be precise about the exact nature of the quantities that appear in the infinite-dimensional case, particularly in the expression for the derivative and Hessian of the objective.

2.1 Notation

Given a Banach space X, we will denote its dual, the space of all bounded linear functionals on X, by X'. We will denote the duality pairing between $T \in X'$ and $v \in X$ by $Tv = \langle T,\ v \rangle$, or by $\langle T,\ v \rangle_X$ if it is desirable to note the space involved. If X is an inner product space, we will denote by $(\cdot\ ,\ \cdot)$ or $(\cdot\ ,\ \cdot)_X$ the inner product. Given two spaces X and Y, $L(X,Y)$ will denote the space of bounded linear maps from X to Y. We will denote by I_X the identity operator on X.

The adjoint of a bounded linear operator $A : X \to Y$ will be denoted by $A^\times$. The adjoint $A^\times : Y' \to X'$ is given by

$$\left\langle A^\times y',\ x \right\rangle_X = \langle y',\ Ax \rangle_Y\ ,\quad y' \in Y'.$$

If X and Y are both Hilbert spaces, we will identify $A^\times$ with the Hilbert space adjoint $A^* : Y \to X$, defined by

$$(x\ ,\ A^* y)_X = (Ax\ ,\ y)_Y$$

for all $x \in X$ and $y \in Y$.

Given a map $G : X \to Y$, we will sometimes denote its first and second derivatives at x by $DG(x)$ and $D^2G(x)$. In the proof of Theorem 2.2 we will need to distinguish between the dependence of the derivatives DG and D^2G on x and their action on vectors, which we will do by using brackets to delimit the arguments of DG and D^2G as linear and bilinear maps: $DG[v] = DG(x)[v]$ and $D^2G[v_1, v_2] = D^2G(x)[v_1, v_2]$.

2.2 The implicit function theorem and implicit differentiation

The classical implicit function theorem [14] will suffice for the calculation of sensitivities in this paper:

THEOREM 2.1 (THE IMPLICIT FUNCTION THEOREM). *Let X, U, and V be Banach spaces, and suppose h is a mapping from an open subset S of $X \times U$ into V. Suppose (a_0, u_0) is a point in S such that*

1. *$h(a_0, u_0) = 0$,*
2. *h is continuously Fréchet differentiable at (a_0, u_0), and*
3. *the partial Fréchet derivative $\partial h/\partial u(a_0, u_0)$ is boundedly invertible.*

Then there exists a neighborhood Σ of a_0 such that for each $a \in \Sigma$, the equation $h(a, u) = 0$ is solvable for $u(a) \in U$. Moreover, the derivative of this solution $u(a)$ with respect to a is given by

$$\frac{du}{da} = -\left(\frac{\partial h}{\partial u}\right)^{-1} \frac{\partial h}{\partial a}. \tag{5}$$

This formula for the Jacobian of u with respect to a is formally the result of applying implicit differentiation to $h(a, u(a)) = 0$ to obtain

$$\frac{\partial h}{\partial a} + \frac{\partial h}{\partial u}\frac{du}{da} = 0$$

and thence (5).

2.3 The reduced derivative and the reduced Hessian

We will now apply the Implicit Function Theorem to derive formulae for the derivative and Hessian of the objective function F in (1). We will assume that $u(a)$ is a locally unique solution to

$$h(a, u(a)) = 0, \tag{6}$$

where $h : (a, u) \in X \times U \to V$, and that $\partial h/\partial u$ is boundedly invertible. In practice, the validity of these hypotheses typically follows from the existence and uniqueness theory for the solution of the equation represented by (6). We will also suppose that f and h are twice continuously Fréchet differentiable on a neighborhood of $(a, u(a))$.

Let

$$W = W(a, u) = \begin{pmatrix} I_X \\ \dfrac{du}{da} \end{pmatrix} = \begin{pmatrix} I_X \\ -\left(\dfrac{\partial h}{\partial u}\right)^{-1} \dfrac{\partial h}{\partial a} \end{pmatrix}. \tag{7}$$

We will call W the injection operator since it is a one-to-one mapping from X into $X \times U$ and is invertible on its range; in finite dimensions it is a full rank

matrix. Its adjoint $W^\times$ we will call the reduction operator. Observe that the range of W lies in the nullspace of the Jacobian of h:

$$D_{(a,u)}h \;\; W = \left(\frac{\partial h}{\partial a}, \quad \frac{\partial h}{\partial u}\right)\left(\begin{array}{c} I_X \\ -\left(\dfrac{\partial h}{\partial u}\right)^{-1}\dfrac{\partial h}{\partial a} \end{array}\right) = 0. \tag{8}$$

Also define $\lambda \in V'$ by

$$\lambda = -\frac{\partial f}{\partial u}\left(\frac{\partial h}{\partial u}\right)^{-1} \tag{9}$$

and the Lagrangian $\ell(a, u; \lambda)$ by

$$\ell(a, u; \lambda) = f(a, u) + \langle \lambda, \; h(a, u)\rangle_V \, .$$

The Lagrangian is normally associated with constrained optimization, a point to which we will return in §5, where we will discuss the nature of λ as a Lagrange multiplier estimate known as the *costate* or *adjoint state*.

THEOREM 2.2. *The derivative of F with respect to a is given by*

$$F'(a) = \frac{\partial f}{\partial a} - \frac{\partial f}{\partial u}\left(\frac{\partial h}{\partial u}\right)^{-1}\frac{\partial h}{\partial a} \;\; \Big|_{(a,u(a))}, \tag{10}$$

which may also be written as

$$F'(a) = D_{(a,u)}f \;\; W \;\; \Big|_{(a,u(a))} = D_{(a,u)}\ell(a, u; \lambda) \;\; W \;\; \Big|_{(a,u(a))}, \tag{11}$$

where $\lambda = \lambda(a, u(a))$. The Hessian of F is given by

$$\nabla_a^2 F(a) = W^\times \left(\nabla^2_{(a,u)}\ell((a, u(a); \lambda)\right) W \;\; \Big|_{(a,u(a))}, \tag{12}$$

where

$$\nabla^2_{(a,u)}\ell((a, u; \lambda) = \nabla^2_{(a,u)}f(a, u) + \left\langle \lambda, \; D^2_{(a,u)}h(a, u)\right\rangle_V \, .$$

The term $\left\langle \lambda, \; D^2_{(a,u)}h\right\rangle_V$ warrants explanation. Since $D^2_{(a,u)}h(a, u)[v_1, v_2] \in V$ for $v_1, v_2 \in X \times U$, we have a real-valued bilinear form defined by

$$\left\langle \lambda, \; \nabla^2_{(a,u)}h\right\rangle_V [v_1, v_2] = \left\langle \lambda, \; D^2_{(a,u)}h[v_1, v_2]\right\rangle_V \, .$$

In the finite-dimensional case, $h = (h_1, \ldots, h_m)^T$ and we have the more recognizable quantity

$$\left\langle \lambda, \; D^2_{(a,u)}h\right\rangle = \sum_{i=1}^{m} \lambda_i \nabla^2_{(a,u)}h_i.$$

Theorem 2.2 reduces to familiar results from nonlinear programming in the finite-dimensional case. Assuming vectors in $\mathbb{R}^n$ to be column vectors, formula (10) in Theorem 2.2 is an expression for a row vector (a linear functional on $\mathbb{R}^n$). We transpose to obtain the gradient:

$$\nabla_a F = W^T \nabla_{(a,u)} \ell.$$

The objective $F(a) = f(a, u(a))$ is called the *reduced objective*; we obtain the gradient $\nabla_a F$ of the reduced objective by applying the reduction matrix W^T to $\nabla_{(a,u)} f$. This is an instance of the *reduced gradient* in nonlinear programming [11]. For this reason we will call dF/da the *reduced derivative*. Similarly, the expression (12) corresponds to the *reduced Hessian*:

$$\nabla_a^2 F = W^T \, \nabla_{(a,u)}^2 \ell \, W.$$

The reduced gradient and the reduced Hessian and the origin of the terminology "reduced" will be discussed further in §5.

The proof of Theorem 2.2 is a straightforward calculation based on implicit differentiation. The one subtlety is the interpretation of some of the quantities encountered along the way in order to arrive at (12), which looks like the familiar formula for the reduced Hessian. For instance, $\nabla^2 F = W^\times \, \nabla^2 \ell \, W$ means that

$$\nabla^2 F[\eta_1, \eta_2] = \nabla^2 \ell[W\eta_1, W\eta_2] = \nabla^2 \ell[(\eta_1, \frac{du}{da}\eta_1), (\eta_2, \frac{du}{da}\eta_2)]$$

for all $\eta_1, \eta_2 \in X$. The identification of this latter formula with (12) requires the results in §11.

Proof. Computing the derivative of F, we see that

$$\frac{dF}{da}(a) = \frac{\partial f}{\partial a}(a, u(a)) + \frac{\partial f}{\partial u}(a, u(a)) \frac{du}{da}(a).$$

From this and the Implicit Function Theorem we obtain the following expression for the derivative of F:

$$\frac{dF}{da}(a) = \frac{\partial f}{\partial a}(a, u(a)) - \frac{\partial f}{\partial u}(a, u(a)) \left(\frac{\partial h}{\partial u}(a, u(a)) \right)^{-1} \frac{\partial h}{\partial a}(a, u(a)),$$

which is (10). This can be rewritten as

$$\frac{dF}{da}(a) = D_{(a,u)} f \;\; W = \left(\frac{\partial f}{\partial a}, \quad \frac{\partial f}{\partial u} \right) \;\; W;$$

this and (8) yield (11).

We now turn our attention to the Hessian. We have

$$\frac{d^2 F}{da^2} = \frac{d}{da} \left[f_a(a, u(a)) + f_u(a, u(a)) \frac{du}{da} \right],$$

in the sense that for all $\eta_1, \eta_2 \in X$,

$$\frac{d^2F}{da^2}(a)[\eta_1,\eta_2] = \frac{\partial^2 f}{\partial a^2}[\eta_1,\eta_2] + \frac{\partial f_a}{\partial u}[\frac{du}{da}\eta_1,\eta_2] + \frac{\partial f_u}{\partial a}[\eta_1,\frac{du}{da}\eta_2] + \frac{\partial^2 f}{\partial u^2}[\frac{du}{da}\eta_1,\frac{du}{da}\eta_2] + f_u\frac{d^2u}{da^2}[\eta_1,\eta_2],$$

where the partial derivatives on the right-hand side are evaluated at $(a, u(a))$. Here we are using the identification of Hessians and bilinear maps in §11.2. Using the interpretation of adjoints and bilinear forms in (54) in §11.4, we can rewrite this as

$$\begin{aligned}\frac{d^2F}{da^2} &= \begin{pmatrix} I & \frac{du}{da}^\times \end{pmatrix} \begin{pmatrix} \frac{\partial^2 f}{\partial a^2} & \frac{\partial^2 f}{\partial u \partial a} \\ \frac{\partial^2 f}{\partial a \partial u} & \frac{\partial^2 f}{\partial u^2} \end{pmatrix} \begin{pmatrix} I \\ \frac{du}{da} \end{pmatrix} + \frac{\partial f}{\partial u}\frac{d^2u}{da^2} \\ &= W^\times \left(\nabla^2_{(a,u)} f\right) W + \frac{\partial f}{\partial u}\frac{d^2u}{da^2}. \end{aligned} \tag{13}$$

Meanwhile, implicit differentiation of

$$h_a(a,u(a)) + h_u(a,u(a))\frac{du}{da}(a) = 0$$

yields

$$\frac{d^2u}{da^2}(a)[\eta_1,\eta_2] = -\left(\frac{\partial h}{\partial u}\right)^{-1}\left(\frac{\partial^2 h}{\partial a^2}[\eta_1,\eta_2] + \frac{\partial h_a}{\partial u}[\frac{du}{da}\eta_1,\eta_2] + \frac{\partial h_u}{\partial a}[\eta_1,\frac{du}{da}\eta_2] + \frac{\partial^2 h}{\partial u^2}[\frac{du}{da}\eta_1,\frac{du}{da}\eta_2]\right)$$

for all $\eta_1, \eta_2 \in X$, so

$$\begin{aligned}&\frac{\partial f}{\partial u}\frac{d^2u}{da^2}[\eta_1,\eta_2] \\ &\quad = \lambda\left(\frac{\partial^2 h}{\partial a^2}[\eta_1,\eta_2] + \frac{\partial h_a}{\partial u}[\frac{du}{da}\eta_1,\eta_2] + \frac{\partial h_u}{\partial a}[\eta_1,\frac{du}{da}\eta_2] + \frac{\partial^2 h}{\partial u^2}[\frac{du}{da}\eta_1,\frac{du}{da}\eta_2]\right) \\ &\quad = \left\langle \lambda,\ D^2_{(a,u)}h \right\rangle_V [W\eta_1, W\eta_2]. \end{aligned}$$

Since the right-hand side is a real-valued bilinear map, we may again apply (54) in §11.4 to rewrite this as

$$\frac{\partial f}{\partial u}\frac{d^2u}{da^2}[\eta_1,\eta_2] = \left(W^\times \left\langle \lambda,\ D^2_{(a,u)}h \right\rangle W\right)[\eta_1,\eta_2]. \tag{14}$$

Combining (13) and (14) yields (12). □

3 Example

We will apply Theorem 2.2 to compute the derivative for a least-squares functional associated with the following boundary value problem (BVP):

$$\begin{array}{ll} -\nabla \cdot (a \nabla u) + b_i \partial_{x_i} u = q & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{array} \tag{15}$$

We assume Ω is smoothly bounded. We use the summation convention throughout; if an index occurs twice in a quantity then summation over that index is implied: $b_i \partial_{x_i} u = \sum_{i=1}^{n} b_i \partial_{x_i} u$. For simplicity, we will assume that $a = a(x)$ is a scalar function. We will assume, too, that $b_i, q \in L^\infty$. Existence, uniqueness, and regularity of solutions of this problem are discussed in [10, 17].

For simplicity, we have chosen a problem for which the state equation is linear in the state and the boundary values are homogeneous. We will consider the following nonlinear least-squares functional:

$$\text{minimize} \quad F(a) = \tfrac{1}{2} \int_\Omega dx \ (u(x) - u_*(x))^2 ,$$

where $u_* \in L^\infty$. For instance, this objective might represent a parameter estimation problem, in which case the data u_* would represent observations the mismatch with which we wish to minimize. For a further discussion of the parameter estimation problem, see [3, 15, 26] and the references therein. This functional could also arise in inverse design, where u_* would represent some desired state that we are attempting to achieve by varying a. Our goal here is only to study how one computes derivatives, and we will ignore the question of the existence of solutions to the minimization problem.

We will consider weak solutions to (15). For now we will let $X = L^\infty(\Omega)$, though later we also consider the case where $X = C^{k,\alpha}$, the space of C^k functions with Hölder continuous derivatives of order α. A suitable domain for a is

$$S = \{ \, a \in X \ | \ a \geq a_* > 0 \, \}$$

for some positive $a_* \in I\!R$. The state u resides in $U = H_0^1(\Omega)$.

The weak interpretation of the BVP (15) means that the state constraint h is a map

$$h : (a, u) \in S \times U \to h(a, u) \in V = \left(H_0^1(\Omega)\right)'$$

where for $v \in H_0^1(\Omega)$,

$$\langle h(a, u), \ v \rangle_{H_0^1} = \int_\Omega dx \ a \nabla u \cdot \nabla v + \int_\Omega dx \ (b_i \partial_{x_i} u) v - \int_\Omega dx \ qv. \tag{16}$$

The relation that defines u as a function of a is $h(a, u(a)) = 0$ in $\left(H_0^1(\Omega)\right)'$.

We begin by computing the various quantities needed to apply Theorem 2.2. Since h is an affine function in u, it is Fréchet differentiable with respect to u. Computing

$$\frac{\partial h}{\partial u}\nu = \lim_{t\to 0}\frac{h(a,u+t\nu)-h(a,u)}{t}$$

we find that

$$\frac{\partial h}{\partial u}\nu = -\nabla\cdot(a\nabla\nu) + b_i\partial_{x_i}\nu \tag{17}$$

in $\left(H_0^1(\Omega)\right)'$, in the sense that

$$\left\langle \frac{\partial h}{\partial u}\nu,\ v\right\rangle_{H_0^1} = \int_\Omega dx\ a\nabla\nu\cdot\nabla v + \int_\Omega dx\ (b_i\partial_{x_i}\nu)v.$$

In a similar way we obtain

$$\frac{\partial h}{\partial a}\eta = -\nabla\cdot(\eta\nabla u). \tag{18}$$

Again, this equality is to be interpreted in the weak sense, as elements of $\left(H_0^1(\Omega)\right)'$.

Both (17) and (18) are expressions for a Jacobian-vector product—a directional derivative—rather than an explicit formula for the Jacobian. Directional derivatives such as these are straightforward to compute.

Following the program in §2, we wish to apply implicit differentiation. First we check that $\partial h/\partial u$ is boundedly invertible, that is, that for all $\Phi\in\left(H_0^1(\Omega)\right)'$, there exists a weak solution $\nu\in H_0^1(\Omega)$ of the linearized boundary-value problem

$$\begin{aligned}\frac{\partial h}{\partial u}\nu = -\nabla\cdot(a\nabla\nu) + b_i\partial_{x_i}\nu = \Phi \quad &\text{in } \Omega\\ \nu = 0 \quad &\text{on } \partial\Omega,\end{aligned}$$

and that the solution operator is bounded: there exists C, independent of Φ, for which

$$\|\,\nu\,\|_{H_0^1(\Omega)} \le C\,\|\,\Phi\,\|_{\left(H_0^1(\Omega)\right)'}.$$

In this case, the bounded invertibility of $\partial h/\partial u$ follows from the existence theory for elliptic equations in divergence form [10, 25].

Thus we may apply the Implicit Function Theorem to conclude that the action of du/da—the Jacobian of u with respect to a—on a vector η is given by the solution of the linearized BVP

$$\begin{aligned}L\nu = -\nabla\cdot(a\nabla\nu) + b_i\partial_{x_i}\nu = \nabla\cdot(\eta\nabla u) \quad &\text{in } \Omega\\ \nu = 0 \quad &\text{on } \partial\Omega.\end{aligned} \tag{19}$$

This corresponds to

$$\frac{\partial h}{\partial u}\nu = -\frac{\partial h}{\partial a}\eta \quad \text{or} \quad \nu = -\frac{\partial h}{\partial u}^{-1}\frac{\partial h}{\partial a}\eta = \frac{du}{da}\eta$$

in the notation of §2.

We now arrive at the action of the derivative $F'(a)$ on η. Let

$$\nu = \frac{du}{da}\eta;$$

ν is defined by (19). We also have

$$\frac{\partial f}{\partial a} = 0, \quad \frac{\partial f}{\partial u}\nu = \int_\Omega dx\ (u - u_*)\nu.$$

Then by (10), we have

$$F'(a)\eta = \int_\Omega dx\ (u - u_*)\nu. \tag{20}$$

This yields the action of $F'(a)$ as a linear functional.

4 Analytical vs. finite-difference approximation of sensitivities

In this section we will draw some comparisons between the numerical accuracy of the analytical derivatives of §2 and that of finite-difference estimates. We will consider the case where the state equation is linear in u:

$$h(a, u) = A(a)u - b = 0.$$

Given $a = (a_1, \cdots, a_n)$, we compute the matrix $A(a)$ and solve the system $Au = b$ for $u(a)$. For instance, such a linear system would arise in the solution of a boundary-value problem such as (3) or (15). As we shall see, the error estimates are guided by the fact that small changes in a will generally cause only small changes in A, but, if the system is ill-conditioned, may cause much larger changes in u.

Let's see what might happen if we apply finite-differences to compute the partial derivative

$$u'(a) \equiv \frac{\partial u}{\partial a_i}(a),$$

which is the i^{th} column of the Jacobian of u with respect to a.

We will need the following basic estimate concerning the sensitivity of the solution of linear systems to changes in the data, adapted from [13]. Let $\kappa(A)$ denote the condition number of A.

THEOREM 4.1. *Suppose $A \in \mathbb{R}^{n\times n}$ is nonsingular, $b \in \mathbb{R}^n$, $Ax = b$, and suppose $(A + \Delta A)y = b + \Delta b$, where $\| A^{-1} \| \| \Delta A \| < 1$. Then*

$$\frac{\| x - y \|}{\| x \|} \leq \frac{1}{1 - \| A^{-1} \| \| \Delta A \|} \left(\frac{\| A^{-1} \| \| \Delta b \|}{\| x \|} + \| A^{-1} \| \| \Delta A \| \right). \tag{21}$$

Moreover, if $\| \Delta A \| \leq \varepsilon \| A \|$ *and* $\| \Delta b \| \leq \varepsilon \| b \|$, *there are perturbations for which this bound is achieved to first order in* ε.

Of course, this bound is quite pessimistic for most perturbations. For instance, a small perturbation of the form $\Delta A = \alpha A$ is benign, and its effect does not involve $\kappa(A)$. On the other hand, there are perturbations for which these bounds are nearly obtained, which is of significance to us. Moreover, if A has a certain sparsity pattern—say, if A were associated with a finite-difference or finite-element scheme—the perturbations ΔA that produce this sensitivity can have the same sparsity pattern as A.

Let e_i be the i^{th} standard basis vector. We will assume that $a_i \approx 1$, and consider the effect of a finite-difference step $t \approx \tau a_i$, where t reflects the absolute size of the step and τ the relative size. We will use μ to denote machine epsilon, the smallest floating-point number for which $1.0 + \mu = 1.0$ (in floating-point).

Let $u_*(a)$ be the solution to the linear system $A(a)u = b$ computed in exact arithmetic, while $u(a)$ will be the computed solution. Let $e(a) = u(a) - u_*(a)$ be the associated error in the solution; we will assume that u is computed as accurately as possible, so that $\| e(\cdot) \| = O(\kappa(A)\mu)$. We will assume that $\kappa(A)\mu \ll 1$ so we can ignore the issue of numerical singularity.

As we saw in (5), the exact partial derivative $u'_*(a)$ is the solution of

$$A_*(a)u'_*(a) = -\frac{\partial A_*}{\partial a_i}(a)u_*(a), \tag{22}$$

where the subscript '$*$' on the matrices denotes their representation in exact arithmetic. The computed partial derivative $u'(a)$ is the solution of

$$A(a)u'(a) = -\frac{\partial A}{\partial a_i}u(a), \tag{23}$$

where the matrices are the floating-point representations of the exact matrices. Comparing (22) and (23), we expect $\| \Delta A \| = \| A(a) - A_*(a) \| \leq \mu \| A_*(a) \|$, while the change in the right-hand side is

$$\Delta b = \left(\frac{\partial A_*}{\partial a_i} - \frac{\partial A}{\partial a_i} \right) u_*(a) + \frac{\partial A}{\partial a_i} \left(u_*(a) - u(a) \right),$$

from which we obtain

$$\begin{aligned} \| \Delta b \| &\leq \mu \left\| \frac{\partial A_*}{\partial a_i} \right\| \| u_*(a) \| + \left\| \frac{\partial A_*}{\partial a_i} \right\| \| u_*(a) - u(a) \| \\ &\leq \mu(1 + \kappa(A)) \left\| \frac{\partial A_*}{\partial a_i} \right\| \| u_*(a) \|. \end{aligned}$$

We will now make the assumption that

$$\left\| \frac{\partial A_*}{\partial a_i} u_*(a) \right\| \approx \left\| \frac{\partial A_*}{\partial a_i} \right\| \| u_*(a) \| \tag{24}$$

where here $\approx$ means equivalence up to a factor that is small by comparison to $\kappa(A)$. Under this hypothesis, combining the preceding estimates according to (21) we see that computing u' via the analytical formula satisfies a relative error estimate of the form

$$\frac{\| u'_*(a) - u'(a) \|}{\| u'_*(a) \|} = O(\kappa^2(A)\mu). \tag{25}$$

This suggests that computing u' via the analytical formula is comparable in condition to solving least-squares problems. The factor $\kappa^2(A)$ is not entirely unexpected, since the calculation of u' involves the solution of two linear systems, one for u and then another for u'.

Next consider the finite-difference approximation and its two sources of error: truncation error, due to the nonlinearity of the function being differentiated, and condition error, due to inaccuracies in computing the function [11, 20]:

$$\begin{aligned} \frac{u(a + te_i) - u(a)}{t} - u'(a) &= \left(\frac{u_*(a + te_i) - u_*(a)}{t} - u'(a) \right) + \frac{e(a + te_i) - e(a)}{t} \\ &= \text{truncation error} + \text{condition error}. \end{aligned}$$

These are the Scylla and Charybdis of finite-difference approximations, since reducing one error tends to increase the other.

Under our hypotheses, the relative error due to condition error satisfies

$$\frac{\| e(a + te_i) - e(a) \|}{t \| u'_*(a) \|} \approx \frac{\| e(a) \|}{t \| u'_*(a) \|} \leq \frac{\kappa(A)\mu \| u_*(a) \|}{t \| u'_*(a) \|} \approx \frac{\kappa(A)\mu}{\tau} \| A \| / \left\| \frac{\partial A_*}{\partial a_i} \right\|.$$

In practice, condition error is exacerbated by the use of iterative solvers in the solution of the state equations, among other things. In particular, the stopping criteria for iterative methods increases the condition error: consider solving a discretized differential equation, where u would represent a discretized function. The iterative approximation of u might be abandoned when the error in the computed solution is believed to be comparable to the error inherent in the level of the discretization [21], rather then when the relative residual of the system being solved has been reduced to the order of floating-point precision, thus increasing the condition error. However, here we will restrict our attention to the errors solely attributable to the conditioning of the state equations.

Now consider the truncation error. In practice, analytical nonlinearity in u may be amplified by numerical nonlinearity. For instance, numerical methods for the solution of differential equations that contain switches such as upwinding will contribute to the nonlinearity of the dependence of u on a. If we were applying finite-differences to estimating $\partial F / \partial a_i$ in (1) and avoiding the intermediate state u, then we might also have to contend with adaptive meshing methods that could change the state space as a function of a, another contribution to truncation error. Again, we will restrict ourselves here to the effects of the condition of the state equations.

We have

$$\frac{A_*(a+te_i)-A_*(a)}{t}=\frac{\partial A_*}{\partial a_i}(a)+E.$$

We may expect E to be small relative to $A(a)$ if A depends in a straightforward manner on a. For instance, for the example (3), the discretized operator constructed for a finite-difference or finite-element scheme would be a relatively simple algebraic function of the coefficient parameters a. For convenience, define

$$\tilde{u}'_*(a)=\frac{u_*(a+te_i)-u_*(a)}{t}.$$

Then,

$$A_*(a)\tilde{u}'_*(a)=-\left(\frac{\partial A_*}{\partial a_i}(a)+E\right)u_*(a+te_i). \tag{26}$$

Meanwhile, consider $\Delta A=A_*(a+te_i)-A_*(a)$; we expect $\| \Delta A \| \approx \tau \| A_*(a) \|$, and the estimate (21) yields

$$\frac{\| u_*(a+te_i)-u_*(a) \|}{\| u_*(a) \|}\leq\frac{\tau\kappa(A_*(a))}{1-\tau\kappa(A_*)}. \tag{27}$$

Comparing (22) and (26) using the perturbation estimates (21) and (27), we obtain

$$\frac{\| \tilde{u}'_*(a)-u'_*(a) \|}{\| u'_*(a) \|}=O(\kappa^2(A_*(a))\tau).$$

Combining the bounds on the condition and truncation errors, we obtain a bound of the following form on the relative error in the finite-difference estimate:

$$\left\| \frac{u(a+te_i)-u(a)}{t}-u'(a) \right\| / \| u'_*(a) \| \leq c_1\kappa^2(A(a))\tau+\frac{c_2\kappa(A(a))\mu}{\tau}.$$

Minimizing this in τ gives a bound that is $O(\kappa^{3/2}(A)\mu^{1/2})$. In view of our hypothesis $\kappa(A)\mu \ll 1$, this bound is much more pessimistic than the $O(\kappa^2(A)\mu)$ bound on the analytical derivative, itself no great shakes.

This analysis suggests finite-difference approximations of derivatives associated with state equations are potentially much more sensitive to ill-conditioning of the state equations than are derivatives calculated using the analytical formulae. Whether or not one sees these pathologies depends on the condition of the system being solved and the the perturbations of that system caused by changes in the design variables a. And, as we have noted, the analysis sketched here also ignores other sources of error that one encounters in practice that can have an even more pronounced effect.

While in practice one can generally use finite-differences successfully, there remains the possibility for serious and unavoidable errors. One can construct algorithms for unconstrained optimization problems using inexact gradients [5, 22], but errors in the gradient can retard progress. Inaccurate derivatives are

also a problem for sensitivity analysis in design (i.e., approximating the local behavior of a function about a nominal design using a first-order Taylor's series model). The potential for unpredictably inaccurate finite-difference approximations of sensitivities is one motivation for examining analytical techniques for computing derivatives.

5 Relationship of the sensitivity calculations to equality constrained optimization

In §2.3 the Lagrangian

$$\ell(a, u; \lambda) = f(a, u) + \langle \lambda,\ h(a, u) \rangle$$

was introduced with the multiplier $\lambda \in V'$ defined by

$$\lambda = -\frac{\partial f}{\partial u} \left(\frac{\partial h}{\partial u} \right)^{-1}. \tag{28}$$

The motivation for introducing the Lagrangian comes from viewing the problem (4) as an equivalent equality constrained problem:

$$\begin{array}{ll} \text{minimize} & f(a, u) \\ \text{subject to} & h(a, u) = 0, \end{array} \tag{29}$$

where now both a and u are independent variables. From this point of view the costate λ serves as a *Lagrange multiplier estimate* [11, 24]. The assumption that $\partial h/\partial u$ is boundedly invertible allows us to invoke the Karush-Kuhn-Tucker necessary conditions for a feasible point (a_*, u_*) to be a solution of (29) [7]: there exists $\lambda_* \in V'$ for which

$$D_{(a,u)}\ell(a_*, u_*; \lambda_*) = D_{(a,u)}f(a_*, u_*) + \langle \lambda_*,\ D_{(a,u)}h(a_*, u_*) \rangle = 0.$$

In particular, the u-component of this system is

$$\frac{\partial f}{\partial u}(a_*, u_*) + \lambda_* \frac{\partial h}{\partial u}(a_*, u_*) = 0.$$

From this and the definition of the costate (28) we see that λ is an estimate of the Lagrange multiplier associated with (29) that is consistent with the first-order conditions at a locally constrained minimizer: i.e., $\lambda = \lambda_*$ at a minimizer. A further discussion of the topic of multiplier estimates can be found in [11, 24].

The costate λ corresponds to two common multiplier estimates in linear and nonlinear programming, the shadow costs or reduced costs in the simplex method [6] and the variable reduction multiplier estimate in nonlinear programming [11]. To see this correspondence, first consider the Jacobian of the state constraints in the finite-dimensional case:

$$\left(\frac{\partial h}{\partial a},\quad \frac{\partial h}{\partial u} \right) \equiv (N,\quad B).$$

We are assuming that $B = \partial h/\partial u$ is boundedly invertible, so we may take the corresponding variables, the state variables u, as the basic variables (so-called because the columns of B form a basis) and the model parameters a as the nonbasic variables. Then $\lambda^T = B^{-T}\nabla_u f$.

Now consider an iteration of the simplex method for the linear programming problem

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x_L \leq x \leq x_U. \end{aligned}$$

One determines the components x_N of x for which the inequality constraints are binding, and forms an invertible block B from the columns of A corresponding to the remaining components x_B, and a vector c_B from the corresponding components of c. The shadow costs π are then defined to be $\pi = -B^{-T}c_B$, corresponding to the costate λ.

In the case of nonlinear equality constrained programming,

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & h(x) = 0, \end{aligned}$$

the variable reduction multiplier estimate at x is computed by first finding an invertible block of columns B of the Jacobian of h. The multiplier estimate is then $\pi = B^{-T}\nabla_B f(x)$, where $\nabla_B f(x)$ are the corresponding components of the gradient, and again we see the correspondence with λ.

The basic/nonbasic partition comes about by viewing the basic variables as functions of the nonbasic variables. This reduces the problem to one in the nonbasic variables alone; hence "variable reduction," "reduced gradient," and "reduced Hessian." In the case of state constraints, we can treat the state u as a function of a in (29) and eliminate u as an independent variable to obtain (4). The costate multiplier is derived from a fixed partition of the variables in which the state variables are always the basic variables and the model parameters a are always the nonbasic variables. This is unlike the general case of linear and nonlinear programming, in which the basic and nonbasic partition tends to vary.

In the nonlinear programming literature, this relation between equality constrained optimization and systems governed by state relations goes back at least to [1] and work cited there, where it is discussed in the context of the generalized reduced gradients algorithm. Further consequences of the basic/nonbasic partition of the state and model variables can be found in [18].

6 Sensitivity equations vs. adjoint equations

The order of calculation in (5) and (10), which we followed in §3, corresponds to the approach to computing derivatives known as the *sensitivity equations*, as well as computing sensitivities via finite-differences or the forward mode of

automatic differentiation [4]. The sensitivity equations approach is equivalent to computing directional derivatives, and for this reason it is most applicable when there is a small number of design parameters a.

The following example makes the idea clear. We modify our example (15),

$$\begin{aligned} -\nabla \cdot (K_a \nabla u) + b_i \partial_{x_i} u &= q \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

so that the coefficient in the leading term is parameterized as a function of a set of model parameters $a = (a_i)$:

$$K_a = \sum_{i=1}^{n} a_i \phi_i$$

for some (small) set of basis functions $\{\phi_1, \cdots, \phi_n\}$.

Formally, the sensitivity equations are derived by applying $\partial/\partial a_i$ to the governing state equations and interchanging the order of differentiation to obtain a relation defining $\partial u/\partial a_i$:

$$\begin{aligned} -\nabla \cdot (\frac{\partial K_a}{\partial a_i} \nabla u) - \nabla \cdot (K_a \nabla \frac{\partial u}{\partial a_i}) + b_i \partial_{x_i} \frac{\partial u}{\partial a_i} &= 0 \quad \text{in } \Omega \\ \frac{\partial u}{\partial a_i} &= 0 \quad \text{on } \partial\Omega \end{aligned} \tag{30}$$

In terms of the discussion in §§2-3, this is nothing other than implicit differentiation of $h(a, u(a)) = 0$ to obtain

$$\frac{\partial h}{\partial a_i} + \frac{\partial h}{\partial u} \frac{\partial u}{\partial a_i} = 0.$$

The sensitivity equations yield $\partial u/\partial a_i$. If we wish to compute $\partial F/\partial a_i$ for some functional $F(a) = f(a, u(a))$, we would use $\partial u/\partial a_i$ and the chain rule.

The sensitivity equations approach is attractive when one has a large number of outputs but only a relatively small number of inputs. Suppose we wish to compute sensitivities not just for a scalar output F, such as the objective in (1), but a vector-valued function $C(a) = c(a, u(a))$, where $c : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$, such as the constraints in (1). The Jacobian of C is given by

$$\frac{dC}{da} = \frac{\partial c}{\partial a} + \frac{\partial c}{\partial u} \frac{du}{da} = \frac{\partial c}{\partial a} - \underbrace{\frac{\partial c}{\partial u}}_{q \times m} \underbrace{\left(\frac{\partial h}{\partial u}\right)^{-1}}_{m \times m} \underbrace{\frac{\partial h}{\partial a}}_{m \times n}. \tag{31}$$

In the sensitivity equations approach, we tacitly compute du/da as an intermediate quantity, which requires n solutions of the sensitivity equation, no matter the number of state variables u or outputs C. We compute an entire column of the Jacobian of C each time we solve the sensitivity equations.

On the other hand, if one has a relatively large number of inputs, the sensitivity equations may not be practical, since every partial derivative requires the solution of the sensitivity equations (i.e., the linearized state equation (30)). This motivates the *adjoint approach*.

Transpose (31):

$$\nabla C(a) = \nabla_a c - \underbrace{\frac{\partial h}{\partial a}^T}_{n \times m} \underbrace{\left(\frac{\partial h}{\partial u}\right)^{-T}}_{m \times m} \underbrace{\nabla_u c}_{m \times q}, \tag{32}$$

where ∇C denotes the transpose of the Jacobian. Then we see that this transposed sequence of operations requires q solutions of the transposed linearized state equations (q applications of $(\partial h/\partial u)^{-T}$). If $q \ll n$, this will be preferable to the expense of the sensitivity equations approach. This ordering of operations is the gist of the adjoint approach and the reverse mode of automatic differentiation. In the case of $\mathbb{R}^n$, the adjoint corresponds to the matrix transpose.

For an optimization problem, the adjoint equations approach—ordering the calculation of derivatives as in (32)—is very attractive because one obtains the gradient of the objective F, disirregardless of the number of model parameters a, via a single application of the transposed solution operator $(\partial h/\partial u)^{-T}$. More generally, the effort required to compute sensitivities (say, of constraints) via the adjoint approach grows with the number of outputs rather than with the number of inputs.

The adjoint approach requires us to solve linear systems involving $(\partial h/\partial u)^{-T}$. If we have $\partial h/\partial u$ at hand as a factored matrix this is not all that difficult. However, $\partial h/\partial u$ might not be readily available, say, if $h(a, u(a)) = 0$ is solved via a nonlinear fixed-point iteration, or only the action of $\partial h/\partial u$ is available because systems involving it are solved using an iterative scheme. In either case, implementing $(\partial h/\partial u)^{-T}$ will require a fair bit of effort on the part of the user.

In the finite-dimensional case the sensitivity equations and the adjoint approach are simply two different ways of computing a product of matrices. Depending on the relative dimensions of the matrices, one or the other method will be the more attractive. However, in the infinite-dimensional case, the situation is more subtle. The complication arises in the switch from row vectors to column vectors in the adjoint approach, i.e., the transposition of (31) to obtain (32), the significance of which we will now discuss in greater detail.

7 The representation of derivatives and the adjoint approach

We have seen that the attraction of the adjoint approach in finite-dimensional optimization is that one obtains the gradient of the objective for the cost of solving a single linear system. Abstractly, the derivative F' is a linear functional

on $I\!R^n$, while the gradient—the direction of steepest ascent—is a direction in $I\!R^n$. We can pass between the two because of the identification of $I\!R^n$ and its dual, which does not necessarily generalize to the infinite-dimensional case. The derivative of F described in Theorem 2.2 resides in the dual X', and we cannot necessarily identify X' with X. We can connect the two spaces through the notion of a descent direction—a direction $p \in X$ for which $F'(a)p < 0$. At the very least, such a direction is needed in order to apply a quasi-Newton method. This leads us to a discussion of directions of steepest descent, the representation of linear functionals, and the adjoint equations.

7.1 Directions of steepest descent and the action of the Hessian

First recall the definition of a direction of steepest descent [12]. Suppose X is a normed linear space with norm $\| \cdot \|_X$, and suppose $F : X \to I\!R$ is Fréchet differentiable at a with Fréchet derivative $F'(a) \in X'$. Then the direction of steepest descent with respect to the norm $\| \cdot \|_X$ is a solution of the problem

$$\begin{array}{ll} \text{minimize} & \langle F'(a),\ p \rangle \\ \text{subject to} & \| p \|_X \leq 1, \end{array} \tag{33}$$

provided that a solution to this minimization problem exists. In the case of a reflexive Banach space, we are guaranteed at least one solution to (33) because the unit ball B will be weakly sequentially compact [27]. Given any sequence $\{p_k\}$, $\| p_k \| \leq 1$, for which

$$\lim_{k \to \infty} \langle F'(a),\ p_k \rangle = L \equiv \inf_{\| p \| \leq 1} \langle F'(a),\ p \rangle ,$$

the weak sequential compactness means that we can find a subsequence converging to a point p_* for which $\langle F'(a),\ p_* \rangle = L$.

Note that the direction of steepest descent depends on choice of norm—the direction of steepest descent indicates the direction of greatest decrease in F per unit distance, and the distance depends on the norm. The derivative is a linear functional independent of choice of norm; the direction of steepest descent depends on what one means by "steepest". A short step in the L^2 norm may not be a short step in the H^1 norm, for instance, since an oscillatory function may have a small L^2 norm but a very large H^1 norm. This aspect of the choice of norm has practical bearing on the behavior of optimization algorithms. The choice of norm—the scaling—can have a profound impact on the efficiency of optimization algorithms [8, 11].

A similar concern arises in interpreting the action of the Hessian $H = \nabla^2 F$. The Hessian is an element of the space $L(X, X')$ (§11.2); accordingly, the Hessian-vector product Hp is an element of X', and again we ask how this linear functional can be related to directions in X. As with the direction of

steepest descent, a natural problem to pose in order to represent the Hessian-vector product Hp as an element of X is:

$$\begin{array}{ll} \underset{q \in X}{\text{minimize}} & \langle Hp,\ q \rangle \\ \text{subject to} & \| q \| \leq 1. \end{array} \tag{34}$$

In the case of a Hilbert space, we have $X' \approx X$ and $L(X, X') \approx L(X, X)$, so there is an immediate interpretation of Hp as an element of X. In this case, the solution q of (34) will point in the direction of $-Hp$.

The conjugate gradient algorithm illustrates the preceding discussion. Consider the minimization of the quadratic form

$$q(x) = \frac{1}{2} x^T A x - x^T b,$$

where $A \in \mathbb{R}^{n \times n}$ is symmetric positive definite. Following [9], we can summarize the conjugate gradient algorithm as follows:

$x_0 = 0, \bar{r}_0 = b, k = 1$
while $\bar{r}_{k-1} \neq 0$ {
    get d_k such that $d_k^T \bar{r}_{k-1} \neq 0$
    $x_k = \underset{x \in \text{span}\{p_1, \cdots, p_{k-1}, d_k\}}{\text{argmin}} q(x)$
    $p_k = x_k - x_{k-1}$
    $\bar{r}_k = \bar{r}_{k-1} - A p_k$
    $k = k + 1$
}.

In the un-preconditioned conjugate gradient algorithm, at iteration k we minimize q over the span of the preceding search directions and the direction $d_k = r_{k-1} \equiv b - A x_{k-1} = -\nabla q(x_k)$, corresponding to the usual direction of steepest descent with respect to the ℓ^2 Euclidean norm. On the other hand, if we choose $d_k = M^{-1} r_{k-1}$ for a symmetric positive definite M, we obtain the preconditioned conjugate gradient algorithm. However, note that $M^{-1} r_{k-1}$ lies along the direction of steepest descent with respect to the norm induced by the inner product $(x, y)_M = x^T M y$. Thus, computing a direction of steepest descent with respect to an inner product other than the usual Euclidean inner product leads to the preconditioned conjugate algorithm.

The connection between elements of the dual and directions in the domain given by (33) and (34) also allows us to give a sensible interpretation of the following aspect of the conjugate gradient algorithm. Suppose that A comes from a finite-difference discretization of

$$\begin{array}{ll} -\nabla \cdot (a \nabla u) = q & \text{on } \Omega \\ u = 0 & \text{on } \partial\Omega. \end{array} \tag{35}$$

The matrix $A : \mathbb{R}^n \to \mathbb{R}^n$ approximates an infinite-dimensional operator $\hat{A}$ that is a map $\hat{A} : H_0^2 \to L^2$ or $\hat{A} : H_0^1 \to (H_0^1)'$. In the finite-dimensional case, we look for x_k in $\text{span}\{p_1, \cdots, p_{k-1}, d_k\}$, where $d_k = b - Ax_k$. But this does not make sense in terms of the underlying infinite-dimensional problem: d_k lies in what should correspond to the range of $\hat{A}$, and the range and domain of $\hat{A}$ are *not* the same in this case. We can resolve this apparent inconsistency if we view d_k as the solution of a steepest descent problem (33).

7.2 The adjoint approach

The adjoint approach is an approach to computing a direction of steepest descent. The point of view that we present here is that the adjoint approach is a no-holds-barred attempt to express the action of the derivative $F'(a)$ in the following form: For some function $g = g(a)$,

$$\langle F'(a),\ p\rangle = \int gp. \tag{36}$$

The goal of the adjoint approach is to find such a representation, if it exists.

One reason such a representation of the derivative is convenient is that it suggests a direction of steepest descent and a choice of norm (scaling). If, for instance, $g(a) \in X$ and $X \subset L^2$, then $g(a)$ determines the direction of steepest descent in X with respect to the L^2 norm: the Cauchy-Schwarz inequality says that the solution of

$$\begin{array}{ll} \underset{p \in X}{\text{minimize}} & \displaystyle\int gp \\ \text{subject to} & \| p \|_{L^2} \leq 1 \end{array} \tag{37}$$

is $-g/\| g \|_{L^2}$. More importantly, as we will see in §8, a representation of the derivative in the form (36) makes it possible to compute the direction of steepest descent with respect to choices of norm other than the L^2 norm.

Having described the goal of the adjoint approach, we will now give an abstract description of its nature and then pass along to a concrete example. At this point the adjoint equations make their appearance, and we can clarify what is "adjoint" about them.

We start with (10) and play some notational tricks. Given $\eta \in X$,

$$\begin{aligned} F'(a)\eta &= \left(\frac{\partial f}{\partial a} + \frac{\partial f}{\partial u}\frac{du}{da}\right)\eta \\ &= \left\langle 1,\ \frac{\partial f}{\partial a}\eta \right\rangle_{\mathbb{R}} + \left\langle \frac{\partial f}{\partial u},\ \frac{du}{da}\eta \right\rangle_U = \left\langle \frac{\partial f}{\partial a}^{\times} 1,\ \eta \right\rangle_X + \left\langle \frac{du}{da}^{\times}\frac{\partial f}{\partial u},\ \eta \right\rangle_X. \end{aligned}$$

Since

$$\frac{du}{da} = -\left(\frac{\partial h}{\partial u}\right)^{-1}\frac{\partial h}{\partial a},$$

we have

$$\frac{du}{da}^{\times}\frac{\partial f}{\partial u} = -\frac{\partial h}{\partial a}^{\times}\left(\frac{\partial h}{\partial u}\right)^{-\times}\frac{\partial f}{\partial u}. \tag{38}$$

The adjoint equation, represented by $\partial h^{\times}/\partial u$, has now appeared. It is the adjoint of the linearized state relation—adjoint in the sense described in §2.1—and as such always exists.

The solution operator for the adjoint problem is a map

$$\frac{\partial h}{\partial u}^{-\times} : \frac{\partial f}{\partial u} \in U' \to \left(\frac{\partial h}{\partial u}\right)^{-\times}\frac{\partial f}{\partial u} \in V',$$

so

$$-\frac{\partial h}{\partial a}^{\times} : \left(\frac{\partial h}{\partial u}\right)^{-\times}\frac{\partial f}{\partial u} \in V' \mapsto -\frac{\partial h}{\partial a}^{\times}\left(\frac{\partial h}{\partial u}\right)^{-\times}\frac{\partial f}{\partial u} \in X'.$$

This yields the infinite-dimensional analog of (32):

$$F'(a)\eta = \left\langle \frac{\partial f}{\partial a}^{\times} 1 - \frac{\partial h}{\partial a}^{\times}\left(\frac{\partial h}{\partial u}\right)^{-\times}\frac{\partial f}{\partial u},\ \eta \right\rangle_{X}. \tag{39}$$

One hopes that when the dust clears, $F'(a)$ has been revealed in the form (36).

The adjoint approach also leads to an alternative expression for the costate λ. From (9), $\lambda \in V'$ satisfies

$$\left\langle \lambda\frac{\partial h}{\partial u},\ \nu \right\rangle_{U} = -\left\langle \frac{\partial f}{\partial u},\ \nu \right\rangle_{U},$$

for all $\nu \in U$. However,

$$\left\langle \lambda\frac{\partial h}{\partial u},\ \nu \right\rangle_{U} = \left\langle \lambda,\ \frac{\partial h}{\partial u}\nu \right\rangle_{V} = \left\langle \frac{\partial h}{\partial u}^{\times}\lambda,\ \nu \right\rangle_{U},$$

or

$$\lambda = -\left(\frac{\partial h}{\partial u}\right)^{-\times}\frac{\partial f}{\partial u}. \tag{40}$$

allowing us to rewrite (39) as

$$F'(a)\eta = \left\langle \frac{\partial f}{\partial a}^{\times} 1 + \frac{\partial h}{\partial a}^{\times}\lambda,\ \eta \right\rangle_{X}. \tag{41}$$

Also note that the adjoint equations can tell us how to compute an action of the Hessian of F on vectors. If we can identify $p \in X$ with elements of X' through a duality pairing such as (36), and if for all $p \in X$ we can identify

$$\frac{du}{da}^{\times} p = -\frac{\partial h}{\partial a}^{\times}\left(\frac{\partial h}{\partial u}\right)^{-\times} p,$$

which is in X', as an element of X, then the adjoint equations tell us how to compute $W^{\times}$ according to (7), and the action of the Hessian of F via (12).

8 An illustration of the adjoint approach

We will illustrate the adjoint approach using the example introduced in §3. We begin by computing the adjoint equation and the other adjoint operators that appear in (39). We then use these results to compute directions of steepest descent and the action of the Hessian.

8.1 The adjoint equation and other adjoint operators

Recall that $\partial h/\partial u$ maps $v \in H_0^1$ to the linear functional in $(H_0^1)'$ defined by

$$\begin{array}{ll} Lv = -\nabla \cdot (a\nabla v) + b_i \partial_{x_i} v & \text{in } \Omega \\ v = 0 & \text{on } \partial\Omega; \end{array} \tag{42}$$

that is, $(\partial h/\partial u)v \in (H_0^1)'$ is defined by

$$\left\langle \frac{\partial h}{\partial u} v,\ w \right\rangle_{H_0^1} = \int_\Omega dx\ a\nabla w \cdot \nabla v + \int_\Omega dx\ (b_i \partial_{x_i} v) w.$$

for all $w \in H_0^1$.

The adjoint $(\partial h/\partial u)^\times$ maps $w \in (H_0^1)'' \approx H_0^1$ to the linear functional in $(H_0^1)'$ defined by

$$\begin{array}{ll} L^\times w \equiv -\nabla \cdot (a\nabla w) - \partial_{x_i} (b_i w) & \text{in } \Omega \\ w = 0 & \text{on } \partial\Omega. \end{array} \tag{43}$$

To see this adjointness, note that the definition of the adjoint and the reflexive identification of $(H_0^1)''$ and H_0^1 means that

$$\left\langle \frac{\partial h}{\partial u}^\times w,\ v \right\rangle_{H_0^1} \equiv \left\langle w,\ \frac{\partial h}{\partial u} v \right\rangle_{(H_0^1)'} \equiv \left\langle \frac{\partial h}{\partial u} v,\ w \right\rangle_{H_0^1} = \langle Lv,\ w \rangle_{H_0^1}.$$

Meanwhile, the standard weak interpretation of (43) means that for all $w, v \in H_0^1$,

$$\langle L^\times w,\ v \rangle_{H_0^1} = \int_\Omega dx\ a\nabla w \cdot \nabla v + w b_i \partial_{x_i} v = \langle Lv,\ w \rangle_{H_0^1}.$$

Thus (43) defines $(\partial h/\partial u)^\times$.

The operator $(\partial h/\partial u)^{-\times}$ is the solution operator for the boundary value problem (43). Since $(\partial h/\partial u)^{-1}$ is a map $(H_0^1)' \to H_0^1$, its adjoint $(\partial h/\partial u)^{-\times}$ is a map $(H_0^1)' \to (H_0^1)'' \approx H_0^1$, which is again consistent with the interpretation of (43) as representing the weak formulation of a PDE.

We also need to compute $(\partial h/\partial a)^\times$ as part of the adjoint calculation (39). For $\eta \in L^\infty$ we have

$$\frac{\partial h}{\partial a} \eta = -\nabla \cdot (\eta \nabla u) \in (H_0^1)',$$

in the sense that for $v \in H_0^1$ we have

$$\left\langle \frac{\partial h}{\partial a}\eta,\ v \right\rangle_{H_0^1} = \int_\Omega dx\ \eta \nabla u \cdot \nabla v.$$

We have $\nabla u \cdot \nabla v \in (L^\infty)'$; then from

$$\left\langle v,\ \frac{\partial h}{\partial a}\eta \right\rangle_{(H_0^1)'} = \left\langle \frac{\partial h}{\partial a}\eta,\ v \right\rangle_{H_0^1} = \langle \nabla u \cdot \nabla v,\ \eta \rangle_{L^\infty},$$

we see that

$$\frac{\partial h}{\partial a}^{\times} v = \nabla u \cdot \nabla v. \tag{44}$$

Using (43) and (44) we can now compute

$$\frac{du}{da}^{\times} \nu = -\frac{\partial h}{\partial a}^{\times} \left(\frac{\partial h}{\partial u}\right)^{-\times} \nu.$$

We first compute the solution w of

$$\begin{array}{ll} L^{\times} w \equiv -\nabla \cdot (a \nabla w) - \partial_{x_i}(b_i w) = \nu & \text{in } \Omega \\ w = 0 & \text{on } \partial\Omega \end{array} \tag{45}$$

to obtain $w = (\partial h/\partial u)^{-\times} \nu$, and then

$$-\frac{\partial h}{\partial a}^{\times} w = -\nabla u \cdot \nabla w \tag{46}$$

yields $(du/da)^{\times}\nu$.

All these calculations and identifications (rather tediously) work with adjoints in the sense of the definition in §2.1. This sense of adjointness is not that of an inner product space adjoint; the adjointness discussed for this example is certainly not that of a Hilbert space adjoint, for instance. One could attempt to interpret adjointness in this example in terms of the L^2 inner product, but such an interpretation would lead one to unbounded operators on L^2 and significant theoretical complications. The "adjoint" of the adjoint equations should be taken to refer to the adjoint that maps between dual spaces, just as in the theory of weak solutions of differential equations. Thus one avoids unbounded operators. For observations on very similar difficulties with adjoints of unbounded operators to the solution of boundary value problems, see [16].

8.2 Directions of steepest descent

For

$$F(a) = f(a, u(a)) = \frac{1}{2}\int_\Omega dx\ (u - u_*)^2,$$

we have

$$\frac{\partial f}{\partial a} = 0, \quad \frac{\partial f}{\partial u}\nu = \int_\Omega dx\ (u - u_*)\nu. \tag{47}$$

Keep in mind that $\partial f/\partial u = (u - u_*)$ as a linear functional in the sense of (47).

From (40), (43), and (47), the costate $\lambda \in (H_0^1)'' \approx H_0^1$ is the weak solution of

$$\begin{aligned} L^\times \lambda \equiv -\nabla \cdot (a\nabla\lambda) - \partial_{x_i}(b_i\lambda) &= -(u - u_*) \quad \text{in } \Omega \\ \lambda &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{48}$$

The regularity of solutions for the BVP means that we may think of λ as an element of $H_0^1(\Omega)$, but its nature as a Lagrange multiplier in $\left(H_0^1(\Omega)\right)''$ is described via the canonical duality pairing

$$\langle \lambda,\ \Phi \rangle_{(H_0^1)'} = \langle \Phi,\ \lambda \rangle_{H_0^1}, \quad \Phi \in (H_0^1)'$$

that makes H_0^1 isomorphic to $(H_0^1)''$. Here again we encounter the issue of representations of linear functionals.

From (41),

$$F'(a)\eta = \left\langle \frac{\partial h}{\partial a}^\times \lambda,\ \eta \right\rangle_{L^\infty}.$$

Applying (44), we see that if we define

$$g(x) = \nabla\lambda(x) \cdot \nabla u(x) \tag{49}$$

then we arrive at the representation of $F'(a)$ as

$$F'(a)\eta = \int_\Omega dx\ g\eta. \tag{50}$$

This integral representation achieves our first goal in the adjoint approach. This representation will allow us to compute the direction of steepest descent for a variety of norms, as we will now discuss.

At this point the choice of domain X enters our deliberations. Suppose, as we have heretofore, that $a \in X = L^\infty(\Omega)$, and $b_i, q \in L^\infty$. Then we are guaranteed in general only that $u, \lambda \in H_0^1$, and so we can only be assured that the representer g defined in (49) is in L^1. Thus $-g$ does not immediately determine an L^2 direction of steepest descent because we do not know that g is, in fact, in L^2. Without further hypotheses, we cannot simply take the result of applying the adjoint approach as a direction of steepest descent.

However, given that $g \in L^1$, we can compute the direction of steepest descent in the L^∞ norm; it is

$$p(x) = -\text{sign } g(x).$$

Unfortunately, this is not a particularly meaningful direction of steepest descent, and in the computational setting this is not particularly well-scaled. In $\mathbb{R}^n$, the

unit ball in the ℓ^∞ norm contains points with ℓ^2 norm $\sqrt{n}$, so the two norms are quite dissimilar for large n.

One of the problems one can encounter with the adjoint approach has emerged. Even if we can express the derivative in the form (36), the direction of steepest descent suggested by this representation may not be acceptable because of the regularity properties of the representer g.

What happens if we try to improve the regularity of g by restricting attention to coefficients a that are smoother than just L^∞? Well, if $a \in X = \mathbf{C}^\alpha(\bar{\Omega})$ and $b_i, q \in L^\infty$, then $u \in C^{1,\alpha}(\bar{\Omega})$, and $\lambda \in H_0^1$, and so $g \in L^2$. In this case, $p = -g/\| g \|_{L^2}$ would be the direction of steepest descent with respect to the L^2 norm. However, unless $\lambda \in C^{1,\alpha}(\bar{\Omega})$, the direction p may suffer from the flaw that $p \notin X = C^\alpha(\bar{\Omega})$.

It can happen that $\lambda \notin C^{1,\alpha}(\bar{\Omega})$ because the regularity of solutions of the adjoint problem (43) is slightly different from those of the state equation or its linearization, a situation not uncommon in the adjoint approach. In order to guarantee $\lambda \in C^{1,\alpha}$, we must require not only the hypothesis $a \in C^\alpha$ but also $b_i \in C^\alpha$. This is because the differential operator associated with the adjoint contains the weak derivatives $\partial_{x_i}(b_i w)$, terms absent from the operator $\partial h/\partial u$.

Thus, in order to be assured that $\lambda \in C^{1,\alpha}(\bar{\Omega})$, we would need the additional regularity assumptions $b_i \in C^\alpha(\bar{\Omega})$. If these data do not satisfy these conditions, then the L^2 direction of steepest descent defined by (49) is not appropriate. Suppose it were the case that $g \in L^2$ but $g \notin C^\alpha$ and we were to use $p = -g/\| g \|_{L^2}$ in the method of steepest descent, say. If our current iterate a_c were in $X = C^\alpha(\bar{\Omega})$, then immediately we would produce an new iterate $a_+ = a_c + \alpha p$ that is *not* in X. In the computational setting, we could see a marked qualitative change appear in the step from a_c to a_+; possibly "roughness" (oscillations) or features of large magnitude.

However, our difficulties go away if we compute a direction of steepest descent with respect to a higher-order Sobolev norm, say, the H^1 norm. We do this as follows. We seek a solution to the problem

$$\begin{aligned} &\text{minimize} \quad \langle F'(a),\, p\rangle = \int_\Omega dx\; gp \\ &\text{subject to} \quad \| p \|_{H^1} \leq 1. \end{aligned}$$

The Lagrangian for this problem is

$$\ell(p;\mu) = \int_\Omega dx\; gp + \frac{\mu}{2}\int_\Omega dx\; \left(\nabla p \cdot \nabla p + p^2\right),$$

and the first-order necessary condition (which for this convex problem is sufficient) is

$$\frac{d\ell}{dp}(p;\mu)\eta = \int_\Omega dx\; g\eta + \mu\int_\Omega dx\; \left(\nabla p \cdot \nabla \eta + p\eta\right) = 0$$

for all $\eta \in H^1(\Omega)$, with $\mu \geq 0$. But this condition is the same as saying that p is the weak solution of the Neumann problem

$$\begin{aligned} -\nabla \cdot (\nabla p) + p &= -g/\mu \quad \text{in } \Omega \\ \frac{dp}{dn} &= 0 \qquad\qquad \text{on } \partial\Omega, \end{aligned}$$

where $\mu > 0$ is chosen so that $\| p \|_{H^1} = 1$. Thus, in order to compute the direction of steepest descent in the H^1-norm, we first need to compute g as in (49), and then solve this auxiliary Neumann problem. The regularity of solutions of elliptic problems is such that the resulting direction p is not only an element of H^1, but also of $C^{1,\alpha}(\bar{\Omega})$, which is what we wished.

For higher-order Sobolev norms, one would solve the weak form of an auxiliary problem involving a higher-order operator. In this way one can obtain descent directions of ever increasing smoothness, the Sobolev norm acting as a preconditioner. In the computational setting, this would be done using a discrete Sobolev inner product as the weighting for the norm in the optimization algorithm.

8.3 Computing the action of the Hessian

Next we will compute the action of the Hessian and discuss its representation. From (12), $\nabla^2 F = W^\times \left(\nabla^2 \ell\right) W$, meaning

$$\nabla^2 F(\eta_1, \eta_2) = \nabla^2 \ell(W\eta_1, W\eta_2) = \nabla^2 \ell((\eta_1, \frac{du}{da}\eta_1), (\eta_2, \frac{du}{da}\eta_2)).$$

We will see that to compute the action of the Hessian, we must solve two BVP, one of the form (42) and the other of the form (43).

For $i = 1, 2$, let

$$\nu_i = \frac{du}{da}\eta_i.$$

We have

$$\nabla^2 f((\eta_1, \nu_1), (\eta_2, \nu_2)) = \int_\Omega dx\ \nu_1 \nu_2 = \left\langle \frac{du}{da}\eta_1,\ \frac{du}{da}\eta_2 \right\rangle_{H_0^1} = \left\langle \frac{du}{da}^\times \frac{du}{da}\eta_1,\ \eta_2 \right\rangle_{L^\infty},$$

while

$$D^2 h((\eta_1, \nu_1), (\eta_2, \nu_2)) = -\nabla \cdot (\eta_1 \nabla \nu_2) - \nabla \cdot (\eta_2 \nabla \nu_1)$$

in $(H_0^1)'$, and

$$\begin{aligned} &\left\langle \lambda,\ D^2 h((\eta_1, \nu_1), (\eta_2, \nu_2)) \right\rangle_{(H_0^1)'} \\ &\quad = \int_\Omega dx\ \eta_1 \nabla\lambda \cdot \nabla\nu_2 + \int_\Omega dx\ \eta_2 \nabla\lambda \cdot \nabla\nu_1 \end{aligned}$$

$$
\begin{aligned}
&= \langle -\nabla \cdot (\eta_1 \nabla\lambda),\ \nu_2 \rangle_{H_0^1} + \langle \nabla\lambda \cdot \nabla\nu_1,\ \eta_2 \rangle_{L^\infty} \\
&= \left\langle -\nabla \cdot (\eta_1 \nabla\lambda),\ -\frac{du}{da}\eta_2 \right\rangle_{H_0^1} + \langle \nabla\lambda \cdot \nabla\nu_1,\ \eta_2 \rangle_{L^\infty} \\
&= \left\langle \frac{du}{da}^{\times} (\eta_1 \nabla\lambda),\ \eta_2 \right\rangle_{L^\infty} + \langle \nabla\lambda \cdot \nabla\nu_1,\ \eta_2 \rangle_{L^\infty} .
\end{aligned}
$$

Then

$$
\nabla^2 \ell((\eta_1, \nu_1), (\eta_2, \nu_2)) = \int_\Omega dx\ \nu_1\nu_2 + \int_\Omega dx\ \eta_1 \nabla\lambda \cdot \nabla\nu_2 + \int_\Omega dx\ \eta_2 \nabla\lambda \cdot \nabla\nu_1,
$$

or in terms of the various linear functionals,

$$
\nabla^2 \ell((\eta_1, \nu_1), (\eta_2, \nu_2)) = \left\langle \frac{du}{da}^{\times} \frac{du}{da}\eta_1 + \frac{du}{da}^{\times} (\eta_1 \nabla\lambda) + \nabla\lambda \cdot \nabla \frac{du}{da}\eta_1,\ \eta_2 \right\rangle_{L^\infty} .
$$

If we let

$$
\phi = \frac{du}{da}^{\times} \frac{du}{da}\eta_1 + \frac{du}{da}^{\times} (\eta_1 \nabla\lambda) + \nabla\lambda \cdot \nabla \frac{du}{da}\eta_1,
$$

then we see that $\phi \in L^1$ and

$$
\langle \nabla^2 F\ \eta_1,\ \eta_2 \rangle = \int \phi\eta_2. \tag{51}
$$

giving us an integral representation of the action of the Hessian on η_1. As in the case of the representation (50) of the derivative, the choice of domain X and the smoothness of the other data in the problem will determine whether $\phi \in L^2$ or is even more regular.

9 Further observations on the adjoint approach and the representation of the derivative and Hessian

A natural question to ask is when $F'(a)$ can be represented in the form (36). Obviously (36) is natural for a problem posed on L^2, such as many control problems, since then the Riesz Representation Theorem for Hilbert spaces tells us that there exists $g \in L^2$ for which $\langle F'(a),\ p \rangle = (g\ ,\ p)_{L^2}$. However, many problems, such as parameter estimation problems, are not usually posed *a priori* on a Hilbert space such as L^2—there are typically boundedness or regularity constraints on the coefficients in differential operators. So, how common should we expect the representation (36) to be?

The following observation might make us hopeful that the derivative generally can be expressed in the form (36). Suppose the domain X, whatever its

natural topology, is a subset of the Sobolev space H^k for some $k \geq 0$, and the derivative $F'(a)$ is actually a continuous linear functional in the norm of H^k: for some $C > 0$,

$$| \langle F'(a), p \rangle | \leq C \| p \|_{H^k} \tag{52}$$

for all $p \in X$. Using the Hahn-Banach Theorem we can extend $F'(a)$ to a bounded linear functional on all of H^k. We may identify the dual of H^k with the negative norm Sobolev space H^{-k} [2]. This characterization of $(H^k)'$ differs from that given by the Riesz Representation Theorem in terms of the H^k inner product: H^{-k} is defined to be the completion of the space of functionals v on H^k of the form

$$\langle v, p \rangle_{H^{-k}} = \int vp, \quad p \in H^k, \tag{53}$$

for some $v \in L^2$. The completion is taken with respect to the norm

$$\| v \|_{-k} = \sup_{\| p \|_{H^k} \leq 1} | (v , p)_{L^2} | .$$

If (52) holds, then $F'(a) \in H^{-k}$, and since the functionals of the form (53) are dense in H^{-k}, we might hope that we will be able to express $F'(a)$ in the desired form (36), or at the very least approximate it by such simple functionals for which it is trivial to compute a direction of steepest descent. Moreover, functionals of the form (53) are also dense in the duals of other spaces of interest, such as C^k.

Unfortunately, the following elementary proposition points out that our hope for finding a representation of $F'(a)$ of the form (36) and an associated L^2 direction of steepest descent is circumscribed. No cheating is allowed: If one has a representation of $F'(a)$ of the form (36), and this representation is well-behaved in the sense that the representer $g(a)$ can be used to determine an L^2 direction of steepest descent that behaves reasonably as a function of a, then morally the problem can actually be posed on L^2 to begin with.

PROPOSITION 9.1. *Let X and H be Banach spaces such that $X \subset H$. Let S be a convex subset of X and denote by Σ the closure of S in H in the norm on H.*

Also suppose that $F : S \to \mathbb{R}$ is continuously differentiable in the topology of X and that for all $a \in S$ and $\eta \in X$,

$$\langle F'(a), \eta \rangle_X = \langle g(a), \eta \rangle_H ,$$

where $g(a) \in H'$ is bounded in norm as a function of a on subsets of X bounded in the norm on H. Then F extends to a map $F : \Sigma \to \mathbb{R}$ continuous in H.

Proof. For $b, c \in B(0, R) \cap S$ we have

$$F(b) - F(a) = \langle F'(c), b - a \rangle_X = \langle g(c), b - a \rangle_H$$

for some $c \in S$ on the line segment connecting a and b, so

$$| F(b) - F(a) | \leq \| g(c) \|_H \| b - a \|_H \leq K_R \| b - a \|_H$$

where K_R depends only on R. This shows that F is continuous on S in the topology of H, so we can extend F uniquely to a map $F : \Sigma \cap B(0,R) \to I\!R$ continuous in the norm on H. Since $R > 0$ was arbitrary, the proposition follows. □

Suppose that we either express $F'(a)$ as a functional of the form (53), or approximate it by such a functional (as the density of such functionals in many dual spaces might lead us to try). Then Proposition 9.1 says that either F extends to L^2, or the representer $v(a)$ cannot even be bounded in L^2 norm on sets bounded in L^2 norm, much less be continuous. In the latter case, when F does not extend to L^2, the representer produced by the adjoint approach is not by itself a meaningful representation of sensitivities or a direction of steepest descent. In nonlinear programming terms, the descent promised by such a putative direction of descent is not meaningful since the function F is extremely nonlinear with respect to the sense of distance. In the computational setting, this means that the usual direction of steepest descent with respect to the Euclidean norm, i.e., the negative gradient of the discretized problem, may have less and less meaning as the discretization becomes finer.

The conjugate gradient method applied to the BVP (35) in §7 manifests this pathology. The infinite-dimensional operator A does not extend to L^2, so we should not expect a direction of descent computed with respect to the L^2 norm to be useful. The un-preconditioned conjugate gradient algorithm uses approximations of exactly these bad directions of descent and generally does not work well. For a fine discretization, the quadratic form is too nonlinear in the ℓ^2 norm for the ℓ^2 direction of steepest descent to be a useful predictor of the decrease we will see in that search direction.

10 Conclusion

One topic we have not discussed in this paper has been the practical details of the implementation of sensitivity calculations for problem governed by differential equations, particularly the adjoint approach. This is a large topic in its own right, and there is a great deal of disagreement particularly over how the adjoint approach should be implemented. One point of view is to derive the adjoint equations in the infinite-dimensional setting and then discretize them as seen fit. At the other end of the spectrum is the approach that works purely with the discretized problem, and computes the associated derivatives. Automatic differentiation is the extreme of this point of view; not only the discretized state equation but its solution scheme is differentiated. Intermediate to these points of view is one that works with the elements of the discretized problems in ways

that are analogous to how one approaches the infinite-dimensional sensitivity calculation.

Our overview has emphasized the origin of sensitivity calculations in implicit differentiation, and the connection between the sensitivity formulae and variable reduction methods in nonlinear programming. We have stressed the distinction between the derivative and directions of steepest descent as the key to understanding the object and limitations of the adjoint approach. We hope this perspective on the calculation of sensitivities for problems governed by differential equations and other state equations will make discussion easier between nonlinear programmers and those interested in the application of optimization to their specific problems.

The interpretation of the adjoint equations in terms of the Banach space adjoint we have discussed is general. The example of the adjoint approach given in this paper considered a problem involving weak solutions of the governing differential equation, but the ideas apply in the case of classical or strong solutions.

It is not always possible to express the derivative in the form (36). This sometimes occurs, for instance, with objectives F that involve traces of the state u—restrictions of u to lower-dimensional surfaces—because the trace operation makes $\partial f/\partial u$ a distribution. This distribution shows up on the right-hand side of the adjoint problem, and the solution of the adjoint problem may be a distribution that is not a function in the usual sense. In such cases, computing a direction of steepest descent with a norm other than that of L^2, such as the choice of a Sobolev norm discussed in §8.2, will produce a smoother representer for F', which, if sufficiently regular, may serve as a direction of steepest descent.

Computationally, the appearance of a distribution on the right-hand side of the adjoint problem corresponds, say, to taking a function defined on the boundary of a computational grid and injecting it into the interior as a function that is supported only near the boundary. Computing a direction of steepest descent with respect to a Sobolev norm smoothes out this data.

Also note that applying the implicit function theorem to compute derivatives for problems involving traces requires that we know that solutions of the state equation are sufficiently smooth for the trace map to be continuous. An example of a problem for which such trace theorems had to be derived as part of the sensitivity analysis can be found in [19].

One could choose to view the question of norms and scaling that we have discussed as a bogeyman from functional analysis and infinite-dimensional optimization. However, if one is attempting to use approximate a truly infinite-dimensional optimization problem via discretization, then the issue of scaling and the dependence of the direction of steepest descent on the choice of norm will become manifest as the level of discretization increases, as our discussion in connection with the conjugate gradient algorithm indicates. Even when considering the case where the design variables a truly reside in a finite-dimensional domain, one needs to be aware of the issue of scaling. Moreover, when im-

plementing an adjoint approach in either case one will need to understand the nature of the intermediate quantities.

Acknowledgments

The author is very much obliged to Eyal Arian for his careful reading of the draft of this paper; this paper is much improved for his observations. The author also wishes to thank Natalia Alexandrov, Mark LaDue, Stephen Nash, and Virginia Torczon for their helpful comments.

11 Appendix: Some results from operator theory

Relegated to this appendix are some results on operators that are used in connection with the reduced Hessian in Theorem 2.2. These results are identifications that allow us to make the general formula for the reduced Hessian look like the familiar one in $\mathbb{R}^n$.

Given Banach spaces Y, Z, we will denote by $B(Y, Z)$ the space of bounded bilinear maps from Y into Z. Then we have the following equivalences.

11.1 An isomorphism of the space of bilinear maps

There is a natural isomorphism between $L(X, L(U, V))$ and $B(X \times U, V)$, the space of bilinear maps from $X \times U$ into V. Given $A \in L(X, L(U, V))$, we may define a bilinear map $B(x, u) = \langle Ax, u \rangle$. Conversely, given a bilinear map $B : X \times U \to V$, we can define $A \in L(X, L(U, V))$ via $\langle Ax, u \rangle = B(x, u)$.

11.2 Second derivatives as bilinear maps

The derivative of a map $\Phi : Y \to Z$ is a map $D\Phi : y \mapsto D\Phi(y) \in L(Y, Z)$, so its derivative, $D^2\Phi$, is a map $D^2\Phi : y \mapsto D^2\Phi(y) \in L(Y, L(Y, Z))$. Using the identification in §11.1, we may then canonically view $D^2\Phi$ as a bilinear map in $B(Y \times Y, Z)$.

11.3 The adjoint of a bilinear form

A bilinear form B on $\mathbb{R}^n \times \mathbb{R}^m$ has the form $B(x, u) = x^T B u = u^T B^T x$ for some $n \times m$ matrix B. We may view B as mapping $\mathbb{R}^n$ to linear functionals (row vectors) in $(\mathbb{R}^m)'$ via $B : x \mapsto x^T B$, and B^T as mapping $\mathbb{R}^n$ to linear functionals in $(\mathbb{R}^n)'$ via $B^T : u \mapsto u^T B^T$.

The general analog is the following. Suppose that $B_1 : X \times U \to \mathbb{R}$ and $B_2 : U \times X \to \mathbb{R}$ are bounded bilinear forms and that $B_1(x, u) = B_2(u, x)$

for all x, u. Using the identification of §11.1, we have $B_1 \in B(X \times U, \mathbb{R}) \approx L(X, L(U, \mathbb{R})) = L(X, U')$. Likewise, we have $B_2 \in L(U, X')$, and

$$\langle B_1 x,\ u\rangle = \langle B_2 u,\ x\rangle\,.$$

Then $B_1^\times : U'' \to X'$ and $B_2^\times : X'' \to U'$. Since there is a natural embedding $U \subset U''$, we may view $B_1^\times$ as a map $B_1^\times : U \to X'$. Likewise, we may view $B_2^\times$ as a map $B_2^\times : X \to U'$, as desired.

11.4 Composition of linear maps and bilinear forms

Given a bilinear form $B(x, u) = x^T B u = u^T B^T x$ on $\mathbb{R}^n \times \mathbb{R}^m$, then

$$\begin{aligned} B(A_1 x_1, A_2 x_2) &= x_2^T A_2^T B A_1 x_1 = A_2^T B A_1(x_1, x_2) \\ &= x_1^T A_1^T B^T A_2 x_2 = A_1^T B^T A_2(x_2, x_1) \end{aligned}$$

where we are defining the bilinear forms $A_2^T B A_1(x_1, x_2)$ and $A_1^T B^T A_2(x_2, x_1)$ in the obvious way.

The general analog is derived similarly. Suppose that $B : X \times U \to \mathbb{R}$ is a bilinear form, $A_1 : X_1 \to X$, and $A_2 : X_2 \to X$. Then using the interpretation in §11.3 of $B^\times : U \to X'$ we have

$$B(A_1 x_1, A_2 x_2) = \left(A_1^\times B^\times A_2\right)(x_2)(x_1) = \left(A_2^\times B A_1\right)(x_1)(x_2) \qquad (54)$$

References

[1] J. ABADIE, *Application of the GRG algorithm to optimal control problems*, in Integer and Nonlinear Programming, J. Abadie, ed., North-Holland Elsevier, 1970.

[2] R. A. ADAMS, *Sobolev Spaces*, vol. 65 of Pure and Applied Mathematics, Academic Press, 1975.

[3] G. ALESSANDRINI, *On the identification of the leading coefficient of an elliptic equation*, Bolletino U.M.I., Analisi Funzionale e Applicazioni, IV–C (1985).

[4] C. BISCHOF, A. CARLE, G. CORLISS, A. GRIEWANK, AND P. HOVLAND, *ADIFOR: Generating derivative codes from Fortran programs*, Scientific Computing, 1 (1992), pp. 1–29.

[5] R. G. CARTER, *Numerical experience with a class of algorithms for nonlinear optimization using inexact function and gradient information*, SIAM Journal on Scientific Computing, 14 (1993), pp. 368–388.

[6] V. CHVÁTAL, *Linear Programming*, W. H. Freeman and Company, New York, 1983.

[7] K. DEIMLING, *Nonlinear Functional Analysis*, Springer-Verlag, 1985.

[8] J. E. DENNIS, JR. AND R. E. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, 1983.

[9] J. E. DENNIS, JR. AND K. TURNER, *Generalized conjugate directions*, Journal for Linear Algebra and Applications, 88/89 (1987), pp. 187–209.

[10] D. GILBARG AND T. S. TRUDINGER, *Elliptic Partial Differential Equations of Second Order*, Springer-Verlag, second ed., 1983.

[11] P. E. GILL, W. MURRAY, AND M. H. WRIGHT, *Practical Optimization*, Academic Press, 1981.

[12] A. A. GOLDSTEIN, *Constructive Real Analysis*, Harper and Row, New York, 1967.

[13] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[14] L. V. KANTOROVICH AND G. P. AKILOV, *Functional Analysis in Normed Spaces*, International Series of Monographs in Pure and Applied Mathematics, MacMillan, New York, 1964.

[15] R. V. KOHN AND B. D. LOWE, *A variational method for parameter identification*, Mathematical Modelling and Numerical Analysis, 22 (1988).

[16] O. A. LADYZHENSKAYA, *The Boundary Value Problems of Mathematical Physics*, vol. 49 of Applied Mathematical Sciences, Springer-Verlag, 1984.

[17] O. A. LADYZHENSKAYA AND N. N. URAL'TSEVA, *Linear and Quasilinear Elliptic Equations*, Academic Press, 1968.

[18] R. M. LEWIS, *A trust region framework for managing approximation models in engineering optimization*, in Proceedings of the Sixth AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Design, September 1996. AIAA paper 96-4101.

[19] R. M. LEWIS AND W. W. SYMES, *On the relation between the velocity coefficient and boundary value for solutions of the one-dimensional wave equation*, Inverse Problems, 7 (1991), pp. 597–631.

[20] J. N. LYNESS, *Has numerical differentiation a future?*, in Proceedings of the 7th Manitoba Conference on Numerical Mathematics and Computing, D. McCarthy and H. C. Williams, eds., Winnipeg, 1977, Utilitas Mathematica Publishing, pp. 107–129.

[21] S. F. McCormick, ed., *Multigrid Methods*, Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, 1987.

[22] J. J. Moré, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming. The State of the Art, Bonn 1982, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 258–287.

[23] O. Pironneau, *Optimal shape design for elliptic systems*, Springer series in computational physics, Springer-Verlag, 1984.

[24] R. A. Tapia, *Diagonalized multiplier methods and quasi-Newton methods for constrained optimization*, Journal of Optimization Theory and Applications, 22 (1977), pp. 135–194.

[25] N. Trudinger, *Linear elliptic operators with measurable coefficients*, Annali della Scuola Normale Superiore di Pisa, 27 (1973), pp. 265–308.

[26] W. W.-G. Yeh, *Review of parameter estimation procedures in groundwater hydrology: The inverse problem*, Water Resources Review, 22 (1986), pp. 95–108.

[27] K. Yosida, *Functional Analysis*, Springer-Verlag, sixth ed., 1980.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1997	Contractor Report

4. TITLE AND SUBTITLE
A nonlinear programming perspective on sensitivity calculations for systems governed by state equations

5. FUNDING NUMBERS
C NAS1-19480
WU 505-90-52-01

6. AUTHOR(S)
Robert Michael Lewis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Computer Applications in Science and Engineering
Mail Stop 403, NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER
ICASE Report No. 97-12

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-201659
ICASE Report No. 97-12

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Dennis M. Bushnell
Final Report
Submitted to SIAM Review.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified–Unlimited

Subject Category 64

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This paper discusses the calculation of sensitivities, or derivatives, for optimization problems involving systems governed by differential equations and other state relations. The subject is examined from the point of view of nonlinear programming, beginning with the analytical structure of the first and second derivatives associated with such problems and the relation of these derivatives to implicit differentiation and equality constrained optimization. We also outline an error analysis of the analytical formulae and compare the results with similar results for finite-difference estimates of derivatives. We then attend to an investigation of the nature of the adjoint method and the adjoint equations and their relation to directions of steepest descent. We illustrate the points discussed with an optimization problem in which the variables are the coefficients in a differential operator.

14. SUBJECT TERMS
adjoint equations; adjoint method; derivatives; reduced gradient; reduced Hessian; sensitivities; steepest descent

15. NUMBER OF PAGES
37

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		